

An integrated MAC protocol based on DMAC for emergency priority

Zuofei Tan¹, Ren Deng^{2*}, Xiaofu Wei², Chenghao Yu³

¹ Suppression Operation Command Department, Naval Logistics Academy, Tianjin 300450, China

² Logistics Information & Logistic Engineering Department, Army Logistics University of PLA, Chongqing 401331, China

³ Chinese Air Force Jinan Base, Jinan 250002, China

Corresponding Author Email: 975380678@qq.com

<https://doi.org/10.18280/rces.050402>

ABSTRACT

Received: 25 May 2018

Accepted: 11 November 2018

Keywords:

DMAC, emergency, delay, data loss

In a wireless sensor network, the nodes closer to the sink node transmit more data, and consume more energy. This phenomenon lead to more serious network congestion and higher packet loss probability. These series of problems are known as “hot area” effect. Based on the structure of the clusters’ networks in GAF protocol, some researches are carried out on the problems of unfair delay and packet loss. In this paper, a new DMAC protocol-based MAC algorithm is proposed as a supplement to ST-MAC, which is called EDMAC. In this new algorithm, a control information transmission scheduling was added to improve ST-MAC’s data transmission mechanism and a method of emergency priority is provided to ST-MAC to decrease the waiting time of the emergency data within a cycling period.

1. INTRODUCTION

In most applications of wireless sensor network (WSN), the communication distance of the sensor node is limited, while the detection area is far-ranging. To solve this problem, WSN employs the means of multi-hop to transfer data to the sink node, and then to the base station [1-2], as shown in Figure 1. Such a multi-hop process can restrict the sensor network’s energy consumption effectively. But a tree network causes a heap of issues. For example, the more data transferred by a node near the sink node, the more energy consumption and heavier congestion will be caused, which will lead to more packet loss. We call these issues as “hot area” effect [3]. In this paper, we present an algorithm named EDMAC (DMAC in emergency), which ensures that the sensor nodes wake up in emergency, thus equalizing the data delay of the sensor nodes from different layers of a tree network and reducing the data loss rate and network energy consumption. In this paper, EDMAC was expanded to deal with the emergency data and a satisfied result was achieved.

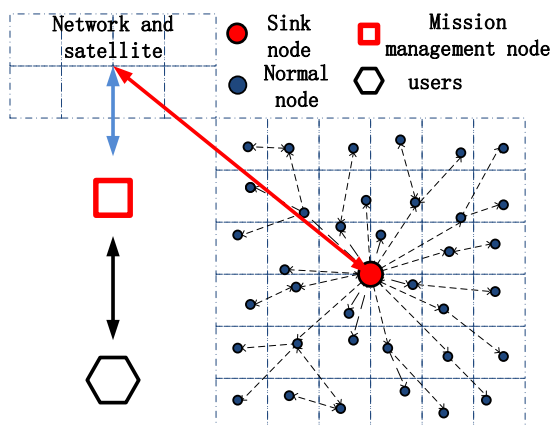


Figure 1. System of wireless sensor networks

Based on the characteristics of tree network and analysis of the network energy problems, USC researchers who summed up predecessors’ experience pointed out that the main cause of the network energy consumption is the sensor nodes’ idle listening [4-7]. Therefore, a gradient MAC protocol named DMAC [8] was taken into account. The wireless sensor network communication can be divided into three parts:

- (1) Data exchange and aggregation between adjacent nodes;
- (2) Control packets obtaining command of interesting incident from the sink node;
- (3) The aggregated data are transmitted from each node to the sink node.

DMAC protocol works in the third part above. This protocol is based on the fast path algorithm (FPA) from SMAC [9] and the mechanism named future request to send (FRTS) from TMAC [10] to avoid the early sleep of the next-hop node. This protocol focuses on a staggered wakeup scheduling mechanism, which makes the data stream flow to the sink node in a given path. However, this protocol has the following defects: (1) It will lead to serious data accumulation, and even overflow around hot spots; (2) Because the protocol is not a fair data transmission strategy [11], it would cause the data entering the cache of the hot spots earlier keep a more front place in the send queue and the data from the node far from the hot spots in the rear of the send queue [12]; (3) In DMAC, there is competition between the nodes at the same layer, which can result in great difference in data delay between nodes in each layer, loss of data, and poor network reliability [13].

In emergency situation, several typical kinds of competitive MAC protocol were used. Such as Ye W’s SMAC, Van Dam T’s TMAC, Hill J’s B-MAC [14] and Zhen Ni’s X-MAC, these competitive protocols can focus on emergency rescue. But in these protocols, most of nodes should keep waked to listen the channel, which would lead to much energy consumption. Another MAC protocols such as RI-MAC [15], PW-MAC [16] both are based on low power probing (LPP)

mechanism, which demonstrates higher throughput and greater power efficiency than B-MAC and X-MAC. But collisions between sending nodes would break up in heavy load using RI-MAC and PW-MAC. In consideration of DMAC's gradual mechanism can avoid collision and conserve energy, an extending based on DMAC is proposed.

2. EDMAC ALGORITHM PRINCIPLE

2.1 Algorithm assumption

- (1) The node location is constant;
- (2) The node energy is enough, and there is no dead node;
- (3) The network structure is constant;
- (4) The initial node transmission path is known;
- (5) Each node keeps synchronization.

2.2 Control information transmission mechanism

When a tree network tries to build the topological relation among sensor nodes, a set of control messages need to be distributed by the sink node. The EDMAC mechanism presents a group of methods to establish the rules for sensor nodes' time slot. This mechanism consists of 3 parts: group control information transmission method, control data packet structure and function, graded control information transmission method.

2.2.1 Group control information transmission method

When the first layer of nodes which are called "hot area" nodes receive the control information from the sink node, as shown in Figure 2, the "hot area" nodes would transmit the control information to the following layer of nodes. In consideration of the broadcast mechanism, a series of data conflict among "hot area" nodes would come up.

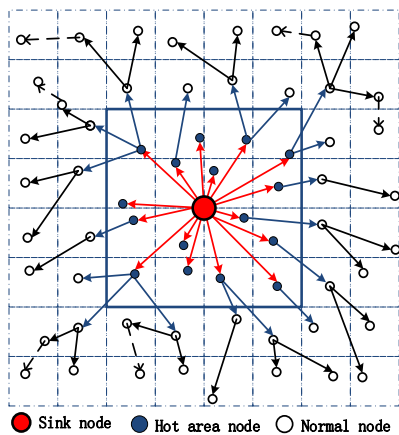


Figure 2. Transmission direction of control messages

So in the EDMAC algorithm, the group transmission method is proposed. In Figure 3, the wakeup slots of "hot area" nodes have been distributed to three adjacent slots by the sink node. And then, the nodes in the following layer would wake up in their leader nodes' wakeup slots to receive control message and send them to their following nodes. According to the DMAC protocol, the length of the control message sending slot is $0.1u$, where u is the length of the slot sending or receiving collected data. In DMAC, nodes in the same layer will compete for the channel during the sending period. So

sending node should wait for a random time back-off period (BP) with a fixed contention window (CW) and receiving node should wait for a short period (SP) and then transmit the ACK to the sending node [17].

2.2.2 Structure and function of control data packet

In wireless sensor network, the sink node needs to adjust the nodes' routes for energy balance, so each layer of a node may change. Then the information of route changing and leader node should be added to the group control packet. So the function of a control packet is to delivery necessary routing, node quantity and time slot information. The group control packet consists of the following parts: updated routing list RL, the number (Nu) of updated neighbor nodes in the same group and the same layer, the ID of the leader node BD, gathering cycle number C_y , and group back-off period tb . In a working gathering network, a cycle means that a data gathering priority in the network has finished and each node's sending or receiving slot would be changed in the next cycle.

The updated routing list RL has 2 functions. Firstly, RL is used to confirm the new routing of receiving nodes; secondly, RL is used to identify the nodes' new layers d . As is shown in Figure 4(a), after node A receives the group control packet from node B, it would update its routing list, which is shown in Figure 4(b). As is shown in this figure, the thick wireframe reveals the layer of the target node. So the target node can broadcast the updating routing information to its child nodes. After this process, node A completes its updating of routing and layer information.

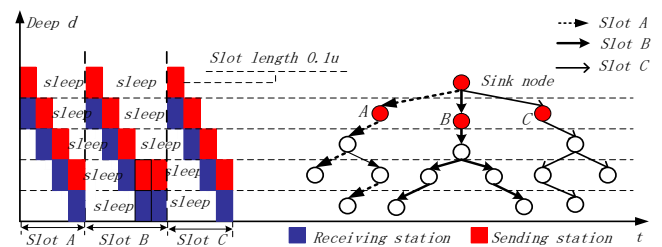


Figure 3. Group transmission mechanism of control messages

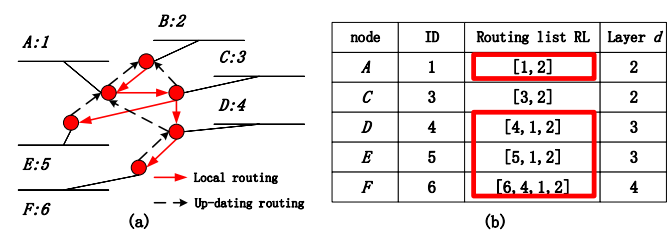


Figure 4. Update function of routing list

Nu is a variable used to calculate the time it takes for each node to receive control signal in the next cycle; BD is the ID of leader node used to distribute the group slot for each node in control packet transmission stage; C_y is the number of the cycle in which each node adjusts its time slot; tb is the group back-off period, a determinant of the sleeping period of each group in the next cycle. As shown in Figure 3, Group B containing of 2 leaf nodes with different parent nodes, calls for another sending time slot. Then, Group C has to wait for the back-off period of the tb information in Group B.

2.2.3 Graded control information transmission mechanism

EDMAC algorithm provides a graded mechanism to transmit control information based on DMAC staggered wakeup schedule. In the control packet transmission stage, each node identifies current routing by the primary routing list (*PRL*), which is shown as the full arrows in Figure 4(a). Each node can get the updated neighbor nodes' number Nu in the same group and the same layer by the *PRL* and its neighbor list. According to the *PRL*, a node can discover whether a neighbor node exists in the same group and in the same layer. If such neighbor node exists, this node would back off a short random period to monitor the busy network before sending control message. The length of the time slot for sending control message satisfies the following condition: $L_{Cr_x} \geq 0.1u$; the length of the time slot for receiving the control information of the lower layer node satisfies the following condition: $L_{Cr_x} \geq 0.1u$. Therefore, the i^{th} group and n th layer node can get the starting time of the time slot for receiving the control message:

$$Cr_{x_i}(n) = \left(1 + \sum_{m=1}^{n-2} Nu_i(m)\right) \times 0.1u + tb_i \quad (1)$$

And the length of the time slot for receiving the control message is:

$$L_{Cr_{x_i}}(n) = Nu_i(n-1) \times 0.1u \quad (2)$$

So it is easy to infer the starting time of the time slot for sending the control message:

$$Cr_{x_i}(n) = Cr_{x_i}(n) + L_{Cr_{x_i}}(n) = \left(1 + \sum_{m=1}^{n-1} Nu_i(m)\right) \times 0.1u + tb_i \quad (3)$$

The length of the time slot for sending the control message is:

$$L_{Cr_{x_i}}(n) = Nu_i(n) \times 0.1u \quad (4)$$

In the formulas above, i is the group ID, n is the layer of the node, tb_i is the length of the back-off duration of the group, u is the duration for receiving and sending the collected data, and:

$$tb_i = \left(\sum_{j=1}^{i-1} N(j)\right) \times 0.1u \quad (5)$$

$N(j)$ is the sum of Nu_j in group j and $L_{max}(j)$ is the deepest layer in group j , so:

$$N(j) = \sum_{m=1}^{L_{max}(j)} Nu_j(m) \quad (6)$$

With these variables in formula 5 and 6, the next group can back off enough period to send its control message. This situation is shown in figure 3, at slot B , there is another node to receive message in group B , and therefore, group C should back off more time at slot C . The back-off period of a group is bound up with its previous group's node number.

2.3 Emergency data transmission mechanism

In DMAC protocol, no emergency treatment scheme is given. When emergency occurs to the gathering sensor node, DMAC protocol still processes the data in the cache of node according to FIFO rule. That is to say the data reaching the cache earlier would have a higher sending priority. Therefore, DMAC protocol will keep emergency data in the cache of node for a long time in a cycle, which will cause severe delay of the emergency data. However, in EDMAC algorithm, commandeering the cycle slot of a front node in the same group. The group is the same in control section. Because of the graded mechanism and priority control, the front node such as the first layer node, may decrease the delay of emergency data.

As is mentioned in the paper before, the wakeup time slot is different in different collection cycles. For this reason, a set of variables should be provided to deduce the formulas of receiving slot and sending slot: TB_h is the duration of the control section in this collection cycle; inside of TB_h , i is the whole groups quantity, for example in figure 3, the quantity of groups is 3, so $i=3$, TB is the total duration of control sections when the network works for several cycles; C_y is the corresponding cycle quantity in these cycles; T is the data gathering cycle; T_c is the longest duration of post-rotation in clusters. In each cycle, all the clusters would rotate to keep energy balance. So:

$$TB_h(j) = \left(\sum_{h=1}^i N(h)\right) \times 0.1u + T_c \quad (7)$$

$$TB = \sum_{j=1}^{C_y} TB_h(j) \quad (8)$$

Substantially, TB is the control message stages' total period in cycles. In the EDMAC algorithm, each node keeps a wakeup list Ci_n of the leader node n , and the emergency wakeup slot list can be denoted in this node as:

$$Tr_m = Ci_n \quad (9)$$

where, m is the node's number. Therefore, the same group nodes' Tr_m s are equal. Due to the graded mechanism, each node in a layer would wake up in the group slot, which can be shown in figure 6. So the node's Tr_m should obey the leader node's Ci_n .

The emergency node m will find the emergency wake-up slot $Tr_m(i)$ which is closest to the incident-occurrence time. Then the node m wakes up at this time slot and requests this slot to send emergency data. If some node in the same layer of the network is going to send cycle data at the same time slot, conflict would occur between the cycle node and emergency node. Therefore, EDMAC sets an emergency-to-send mark (*ETS*) which would be broadcast before an emergency data. And the node at the same layer in the same group also needs to set a corresponding monitor slot at the same time to listen to other *ETS* from the node at the same layer. If the *ETS* mark of the listening nodes is true, the cycle node needs to stop sending cycle data and delete the cycle data in the cache.

Therefore, in the emergency mechanism, each node in the gathering network should set up a monitor slot to listen to *ETS* mark from the node at the same layer. Moreover, if the monitor node is the parent node of the emergency node, the parent node must keep listening at its emergency wakeup slot $Tr_m(i)$ to

receive the emergency data form its child node. To this end, each node from the head to the deepest leaf needs the monitor slot and the listening time is the corresponding element value of emergency wakeup slot Tr_m .

According to the parameter of emergency wakeup slot list Tr_m , the total number of gathering nodes is M , the layer of node n is L_n in the network, the deepest layer is L_{max} and the unit length of time is u . The node m can get the nearest emergency wake-up slot to the emergency time t_{em} , that is to say, t_{em} needs to satisfy the following inequality:

$$t_{em} < T \times (C_y - 1) + TB + [Tr_m(i) - 1] \times 5u + [L_{max} - L_m + 1] \times u \quad (10)$$

When $Tr_m(i)$ satisfies such inequality for the first time:

$$Tr_m(i) > \frac{t_{em} - T \times (C_y - 1) - TB - [L_{max} - L_m + 1] \times u}{5 \times u} + 1 \quad (11)$$

We will take this time slot $Tr_m(i)$ to be the sending slot of emergency. Substituting Formula 1 in literature [18] into Formula 11, we can get:

$$Tr_m(i) > 1 + \{t_{em} - [5 \times u \times M + (L_{max} - 4) \times u] \times (C_y - 1) - TB - [L_{max} - L_m + 1] \times u\} / (5 \times u) \quad (12)$$

So the emergency-occurrence node just needs 5 parameters to calculate the corresponding sending slot of emergency by formula 12.

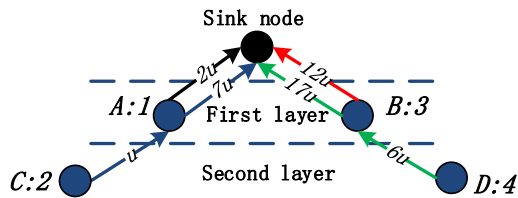


Figure 5. Gathering network of EDMAC

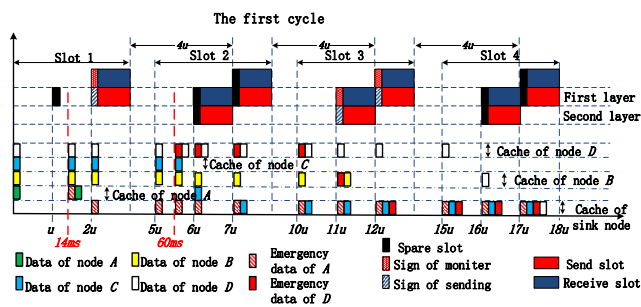


Figure 6. Data transmission mechanism of emergency

Table 1. Wakeup slot time list of emergency

Node ID	A: 1	C: 2	B: 3	D: 4
Wake-up slot list	[1, 2]	[2]	[3, 4]	[4]
Node's slot	[1]	[2]	[3]	[4]
Emergency time	14ms	none	none	60ms
Emergency wake-up slot list	[1, 2]	[1, 2]	[3, 4]	[3, 4]

We take the gathering network of Figure 5 as instance, each node's first cycle emergency-occurrence time t_{em} and the corresponding Tr_m can be shown in Table 1, in which the unit time $u=11ms$ (Huang et al., 2013). So the corresponding network graded data-transmission mechanism can be shown in Figure 6. The node A will wake up to listen to the channel at the time of u and $6u$, and node B will wake up to listen to the channel at $11u$ and $16u$, both of which are going to estimate whether any ETS mark exists, as is shown by the red dotted line. First, node A generates an emergency data at $14ms$ and checks the nearest emergency wakeup slot list Tr_A . Table 1 shows that the nearest emergency wakeup slot to $14ms$ is slot 1. Then node A will broadcast an ETS mark primarily, which is shown as the blue twill at the time of $2u$ in Figure 6. For example, node D generates an emergency data. Because node D 's emergency wakeup slot list just contains slot 3 and slot 4, node D will not wake up to send emergency data at slot 2. But at the same time, node A will wake up to listen to the channel at the time of $6u$, just because node A 's Tr_A contains slot 2, which is shown as a black slot at $6u$ in Figure 6.

The emergency data transmission mechanism can decrease the delay of emergency data. Because the length of emergency wakeup list Tr_m is equal to the number of members in the group of emergency-occurring node. The more member the group has, the more usable time slot the group has. Thus, it can guarantee the equality of the equity among different groups.

3. SIMULATION DESIGN

3.1 Simulation assumption

Our algorithm is simulated via MATLAB and the relevant parameters are shown in Table 2. To simplify the generation of network topology, a network was created in an $200m \times 200m$ monitoring area by random distribution. This network is shown in Figure 7 and we presume that the network structure cannot be changed. In Figure 7, the red connections indicate the hot area nodes, the blue full line connections are second hop nodes to the sink node and the blue imaginary line connections are the third hop nodes.

To illustrate the improvement of the hot spots effect by our algorithm, in this section, 3 sets of performance indicators were established: average end-to-end delay of all nodes in emergency, Packet loss probability in emergency, comparison of energy consumption in emergency mode. These indicators are the common criteria which also were used in literature. By contrast, the EDMAC algorithm has good performance in all of the above aspects.

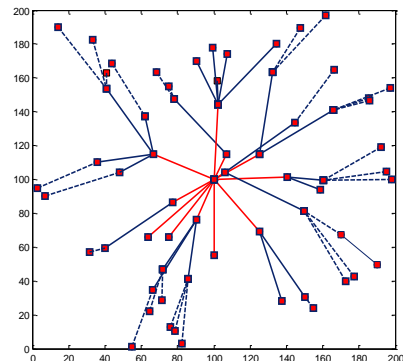


Figure 7. Simulation of topological structure

Table 2. Parameter setting of EDMAC algorithm

Parameter name	value
Area	200m×200m
Location of sink node	(100m, 100m)
Length of sending and receiving time u	11ms
Length of shortest sleeping time	33ms
Length of <i>MTS</i> mark	1ms
Length of emergency monitoring	1ms
Length of <i>ETS</i> mark	1ms
Nodes' number	64
Communication radius	100m
Disturbance radius	200m
Sending power	0.66w
Receiving power	0.39w
Idle power	0.35w
Gathering cycle	3520ms
Total cycles	64

3.2 Simulation analysis

3.2.1 Average end-to-end delay of all nodes in emergency

Figure 8 shows the average end-to-end delay of all nodes in emergency, in which, the x-coordinate is node's ID and the y-coordinate is the end-to-end delay, and furthermore, the unit is *ms*.

As is shown in Figure 8, with the depth difference in DMAC, the emergency delays in nodes are widely different. The reason is its *FIFO* mechanism, the closer nodes to the sink node has a slighter delay. But in EDMAC, a rank has been provided in each node's cache as per the emergencies and a new monitor slot has been added for emergencies. A rank for emergencies can make sure those emergency data always stay at head of node's cache. A monitor slot let each emergency data be sent out at the nearest slot. Those mechanisms efficiently reduce the end-to-end delay.

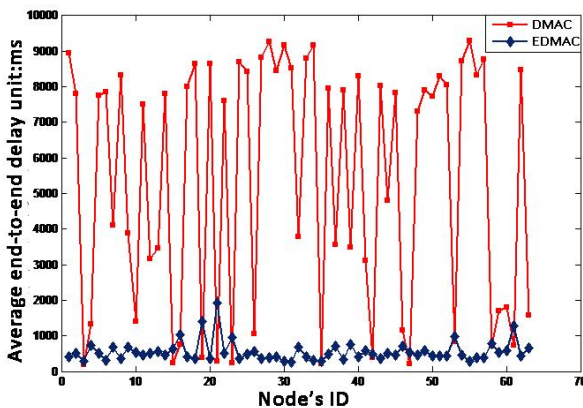


Figure 8. Average end-to-end delay of all nodes' emergency

3.2.2 Packet loss probability in emergency

Figure 9 is the Packet loss probability in emergency, in which, the x-coordinate is the network performance period and the y-coordinate is the packet loss probability of emergency.

In EDMAC, the Emergency wake-up slot list is important. On one hand, the list offers a set of optional emergency slots. On other hand, this list reflects the quantity diversity among

groups' nodes. The more nodes the group includes, the more slots can be used to balance the data-sending opportunity among groups. But in DMAC, in the group with more nodes, the opportunity to send data is the same in the fewer nodes' group. A conflict would break up more easily in a bigger group. So as is shown in figure 9, EDMAC has lower packet loss probability than DMAC.

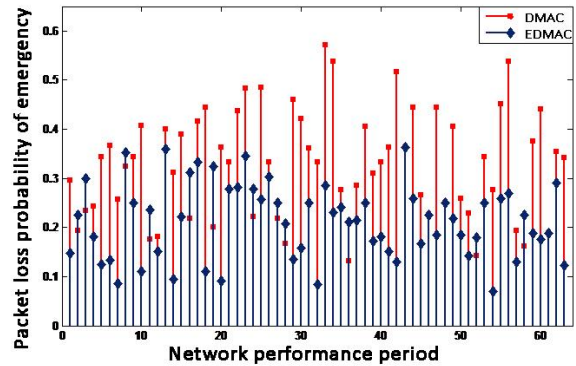


Figure 9. Packet loss probability of emergency

3.2.3 Energy consumption in emergency mode

Figure 10 is the comparison of energy consumption in emergency mode, in which, the x-coordinate is the network performance period and the y-coordinate is the total consumption of different protocols.

EDMAC needs to monitor the channel in each emergency wakeup slot to find the sign of emergency, which is shown in Figure 6, while DMAC would spend more energy in listening in each wake-up slot. The quantity of wakeup slot is completely larger than that of the emergency wake-up slot. The consequence is shown in Figure 10.

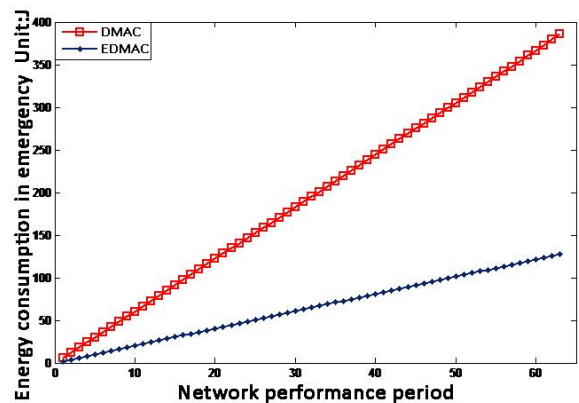


Figure 10. Comparison of energy consumption in emergency mode

4. CONCLUSION

This paper provides an effective algorithm (EDMAC) for emergency data based on DMAC. The protocol is a supplement to ST-MAC. In this new algorithm, a control information transmission schedule is added to improve ST-MAC's data transmission mechanism and a method of emergency priority is provided to the ST-MAC to decrease the waiting time of the emergency data within a cycle period. The consequence indicates that EDMAC contributes to fewer

emergency delay, lower emergency packet loss probability and less energy consumption.

ACKNOWLEDGMENT

This work was supported by the project of Real Estate Management System under No.AS214R002. We thank the anonymous reviewers whose comments helped improve the manuscript.

REFERENCES

- [1] Villas LA, Boukerche A, Ramos HS. (2013). DRINA: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *IEEE Transactions on Computers* 62(4): 676-689.
- [2] Ren F, He T, Das SK. (2011). Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks. *IEEE Transactions on Parallel & Distributed Systems* 22(9): 1585-1599. <https://doi.org/10.1109/TPDS.2011.24>
- [3] Nethravathi NC, Ballal PA. (2015). Energy based-genetically derived secure cluster-based data aggregation in wireless sensor networks. *Plos Neglected Tropical Diseases* 121(13): 1093-1100. <http://dx.doi.org/10.5120/21603-4714>
- [4] Joshi S, Jaiswal AK, Tyagi PK. (2013). A novel analysis of T Mac and S Mac protocol for wireless sensor networks using castalia. *International Journal of Soft Computing & Engineering* 2(6): 128-131.
- [5] Wu SH, Chen MS, Chen CM. (2014). Fully adaptive power saving protocols for Ad Hoc networks using the hyper quorum system. *IEEE/ACM Transactions on Networking* 22(1): 1-15. <http://dx.doi.org/10.1109/ICDCS.2008.88>
- [6] Carrano RC, Passos D, Magalhaes LCS. (2014). Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys & Tutorials* 16(1): 181-194. <https://doi.org/10.1109/SURV.2013.052213.00116>
- [7] Garcia-Saavedra A, Serrano P, Banchs A. (2012). Energy consumption anatomy of 802.11 devices and its implication on modeling and design. 8th ACM International Conference on Emerging Networking Experiments and Technologies, Nice, France, pp. 169-180. <https://doi.org/10.1145/2413176.2413197>
- [8] Lu G, Krishnamachari B, Raghavendra CS. (2007). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. *Wireless Communications & Mobile Computing* 7(7): 863-875. <https://doi.org/10.1109/IPDPS.2004.1303264>
- [9] Huang P, Xiao L, Soltani S, Mutka MW, Xi N. (2013). The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials* 15(1): 101-120. <https://doi.org/10.1109/SURV.2012.040412.00105>
- [10] Singh P, Varma S. (2014). An improved TMAC protocol for wireless sensor networks. *International Conference on Signal Propagation and Computer Technology*, pp. 91-95. <https://doi.org/10.1109/ICSPCT.2014.6884921>
- [11] Sunitha GP, Kumar SMD, Kumar BPV. (2015). A preemptive multiple queue based congestion control for different traffic classes in WSN. *International Conference on Circuits, Communication, Bangalore, India*, pp. 212-218. <https://doi.org/10.1109/CIMCA.2014.7057793>
- [12] Söderman P, Grinnemo KJ, Hidell M, Sjödin P. (2015). Mind the SmartGap: A buffer management algorithm for delay tolerant wireless sensor networks, wireless sensor networks. Springer International Publishing 104-119. https://doi.org/10.1007/978-3-319-15582-1_7
- [13] Zeng SG. (2012). Reliability and lifetime research in wireless sensor network. Ph.D. dissertation. East China University of Science and Technology, Shanghai, China.
- [14] Polastre J, Hill J, Culler D. (2004). Versatile low power media access for wireless sensor networks. *International Conference on Embedded Networked Sensor Systems*. Baltimore, USA, pp. 95-107. <https://doi.org/10.1145/1031495.1031508>
- [15] Sun Y, Gurewitz O, Johnson DB. (2008). RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *International Conference on Embedded Networked Sensor Systems*, Raleigh, USA, pp. 1-14. <https://doi.org/10.1145/1460412.1460414>
- [16] Tang L, Sun Y, Gurewitz O, Johnson DB. (2011). PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. *INFOCOM*: 1305-1313. <https://doi.org/10.1109/INFCOM.2011.5934913>
- [17] Li YX, Shi HS, Pang BM. (2012). An energy-efficient MAC protocol for wireless sensor network. *Lecture Notes in Electrical Engineering* 143(9): 163-170. <http://dx.doi.org/10.1109/ICACTE.2010.5579102>
- [18] Fei ZT, Lin SL, Xiao FW, Xia ZW. (2017). An algorithm of fair delay in MAC layer. *Journal of Logistical Engineering University* 33(1): 85-89.